Int. J. Advanced Networking and Applications 5455
Volume: 14 Issue: 03 Pages: 5455-5463(2022) ISSN: 0975-0290

Improved Parameter-Free 3D Object Retrieval (IP3DOR) System with Hierarchical Clustering

Abubakar Roko
Department of Computer Science, UsmanuDanfodiyo University, Sokoto, NIGERIA
Email: abroko@yahoo.com
Aminu Lawal
Department of Computer Science, UsmanuDanfodiyo University, Sokoto, NIGERIA
Email: Talk2aminulawal@gmail.com
Aminu B. Muhammad
Department of Computer Science, UsmanuDanfodiyo University, Sokoto, NIGERIA
Email: aminu.muhd@yahoo.com
Abdulgafar Usman
Department of Computer Science, UsmanuDanfodiyo University, Sokoto, NIGERIA
Email: abdulgafarusman80@gmail.com

ABSTRACT

In content-based 3D object retrieval, searching for a query object in an extensive database is essential. The existing retrieval algorithms adopt the naïve search algorithm in searching for query objects. This approach leads to a high cost of search and retrieval that needs to be addressed. In this research, we introduced an algorithm that calculates each cluster's representative, that is, a 3D object that has the least dissimilarity on average to each object in the cluster. This is to improve the overall retrieval performance of [1]. We first compute the optimal hierarchical level of the database using a dendrogram, and then calculate the total number of clusters in the database. Afterwards, we calculate the feature descriptor of each cluster. When a user chooses a query object, our system then compares the feature descriptor of the query object with each of the cluster's representation and search the cluster with the smallest distance to the query, thereby improving the query searching and improving the system. The proposed system was implemented, and the system's performance was evaluated against the benchmark datasets. This revealed that the execution time was reduced by 21% and increased Precision and Recall by 30.7% and 33.1%, respectively. In the future, it is suggested that the technique be improved by incorporating different machine learning algorithms and comparing the results.

Keywords **– 3D Object Retrieval, Hierarchical Clustering, Cluster Representative.**

Date of Submission: Oct 13, 2022 Date of Acceptance: Nov 16, 2022

1. INTRODUCTION

Three Dimensional (3D) objects are solid forms like boxes, cones, and balls having three dimensions: length, width, and height. Unlike two-dimensional (2D) shapes, three-dimensional shapes include thickness or depth. 3D objects are increasingly essential since they can show more detailed information than 2D, which is relevant to different fields such as medicine, molecular biology, military application, and entertainment, among others. Because of improved methods in scanning, modeling, and presenting 3D objects, a significant volume of 3D objects is accessible all over the internet. The first experimental 3D Model retrieval systems include the Ephesus search engine at the National Research Council of Canada, Princeton University, National Taiwan University, National Institute of Multimedia Education in Japan (Ogden IV system), and Informatics and Telematics Institute in Greece.3D models' retrieval systems have two main methods of retrieving a 3D model: text-based and content-based methods [2]. Text-based involves mainly using metadata such as keywords or captions to retrieve a 3D object from a database [2]. However, this approach is ineffective for 3D models since it suffers from the drawbacks such as poor efficiency, low accuracy, and considerable uncertainty [2]. Content-based 3D object retrieval entails searching and retrieving a 3D model from a database based on its appearance, color, shape, and texture [3]. In the 3D Model retrieval system, we have a database and a query object. A query object is also a 3D object itself. According to [4], the main task of a 3D model retrieval system is "given a query object, define appropriate measures to automatically assess the similarity between any pair of 3D objects based on a suitable notion of similarity". The main challenge in similarity comparison between two 3D objects is to develop shape descriptors that could construct an index efficiently and accomplish geometric shape matching effectively.[3]. In general, a shape descriptor is a simplified representation of a 3D model in the form of a vector containing a set of numerical values or a graph-like structure used to describe the shape geometrically or topologically. A good shape descriptor should be transformation invariant, user-friendly, high-performance computation, index structures, and ease of storing.[5]. In most 3D object retrieval systems, there are several approaches to querying the system, query by example, Sketch-based query, direct querying, or browsing through the database [6], [7], [8].

The user provides an existing 3D model in the query by example approach. While indirect querying, the user directly gives the query description of the intended 3D model. This technique can only be used if the descriptor is in a readable format. Lastly, the user may browse through the database to query the system. A common feature of all 3D object database systems is that queries look for comparable objects rather than an exact search, such as in conventional relational databases. Three-dimensional objects cannot be effectively searched in the conventional sense (exact search) since the likelihood of two three-dimensional objects being identical is very minimal unless they are electronic copies from the very same source. Rather than that, a 3D database system query often requests objects that are the most comparable to the query item in question or a manually specified query definition. As a result, one of the critical objectives in developing a 3D retrieval system is developing effective and efficient similarity search algorithms.

The primary objective of a 3D similarity search is to develop algorithms that efficiently perform similarity queries in 3D databases. Effectiveness is assessed in recovering comparable 3D objects while retaining dissimilar ones and efficiency is measured in terms of the cost of the search run time, which may be quantified in terms of CPU or I/O time.The efficiency and effectiveness of 3D object retrieval systems remained subpart is research as researchers develop new algorithms daily to increase the efficiency of the existing.

2. RELATED WORKS

2.1 FEATURE-BASED 3D MODEL RETRIEVAL METHODS

[9]Proposed 3D shape retrieval by employing surface moment invariants as feature descriptors of supervised 3D objects. Surface Moment invariant is unaffected by translation, scaling, and rotation transformations. It indicates that the Surface Moment invariant yields the same feature vector regardless of similarity transformation. The descriptor generated 54 distinct moment invariants in the fifth order. A 54- dimensional vector represents each model. Shape retrieval combined with prior knowledge and classification information significantly improves the retrieval result. However, this approach suffers from the additional run time of the learning process. The learning process from the training set by the back-propagation neural network is time-consuming.

[10]Proposed PANORAMA, which uses panoramic views of a three-dimensional object to describe its shape in a novel manner. It does this by projecting it onto a set of cylinders aligned with the object's principal axis. Unsupervised relevance feedback technique (RF) is used to increase retrieval performance. There are three steps to computing PANORAMA: Pose Normalization, Extraction of panoramic views, and feature extraction. Pose Normalization normalizes the 3D object's pose so that translation, rotation, and scaling do not affect the descriptor. In normalizing a 3D object's translation, the centroid is determined using Continuous Primary Component Analysis (CPCA), whereas to normalize its rotation, both CPCA and Normal PCA (NPCA) are performed to align the model's principal axes with the coordinate axes. Finally, on pose normalization, the descriptor is made scale-invariant by normalizing the features that make it up to the unit L1 norm, found in the 2D Discrete Fourier Transform and the 2D Discrete Wavelet Transform. To get panoramic views, you first need to normalize the 3D model. Then, you'll project it onto the sides of cylinders aligned with the object's principal axis. Compared to another cutting-edge descriptor, such as DESIRE [11], PANORAMA provides more outstanding and efficient retrieval performance. However, since the descriptor is augmented with local relevance feedback (LRF) technique, it suffers from query drift. This occurs when the retrieval system is misled by the irrelevant data and drawn away from the user’s target.

[12]Combine ZFDR, a hybrid descriptor, with CBR, a class-based retrieval method that uses class information. The algorithm is called CBR-ZFDR, named after the two methods mentioned above. Features like Zernike moments and Fourier descriptor are part of the FDR and depth information and ray-based features. The Zernike moment and Fourier descriptor features make up the virtual information. The depth and Ray-based feature make up the geometric information. You start the shape description computation by normalizing the object's pose by calculating the object's enclosing sphere. The 3D object is then translated until the enclosing ball and the point of intersection of coordinate axes are the same distance. Finally, the object is scaled until the radius of the enclosing sphere surrounding it is 1; after the pose normalization and descriptor extraction, the similarity between the user's query and every other 3D model in the dataset. Class distance computation is done immediately, and this is where the dissimilarity between the user's query and a class in the dataset is computed. Finally, it is found out how far the two objects are from each other, and the list of items is arranged in order of similarity. In general, the algorithm does better than most of the best descriptors out there. However, the main problem with this method is that it doesn't consider 3D model databases that haven't been categorized by the method yet.

2.2 ALGORITHMS AUGMENTED WITH MACHINE LEARNING TECHNIQUES

[13]Proposed a semi-supervised clustering method based on a support vector machine (SVM) that organizes 3D models on a semantic level and then runs a content-based search from the semantic cluster. This is how supervised learning works: training data is used to figure out the pattern of each semantic category. Then, anonymous data is automatically classified and clustered based on the pattern. The system first learns the pattern for semantic clustering from training data. Then it clusters the database in an unsupervised way based on the pattern is found. Finally, the unified search strategy does a content-based search from the formed semantic clusters. This method combines supervised classification and content-based

retrieval to make it easier to find what you're looking for. However, the major drawback of this approach is that it is possible to misclassify a model, for example, classifying a gear model as a screw model.

There are several feedback techniques that [10] look at to help bridge the semantic gap between low-level features used by the system and high-level semantic information used by humans in 3dor. These techniques include One-Class SVM, Query Modification, Multiple Queries, and pseudo–Relevance Feedback (PRF). People who use Relevance feedback (RF) use low-level features to show a list of results based on how similar they are to a first query. Then, the user looks through the list of results and gives the system information about how relevant a few of them are by giving relevant feedback. The system adjusts its parameters to match the user's classification criteria the best. Then, based on the changed parameters, a new search session is started, and new results are shown to the user. This process is repeated until the user is satisfied. We've talked about three ways to do this: one Class SVM, Query Modification, and Multiple Queries (Mul-Q). Use a set of observations (feature vectors) from the same target class to find a hyper-sphere in the feature space with the most observations while having the smallest radius. As a result of the Q-mod technique, a near-neighbor search can be improved by getting more relevant objects at the top of the list. Finally, in MUL-Q, many queries are run, and the results are combined into a single report. There are three ways to do this: OC-SVM is the least effective; multiple queries are the most effective. Later, they talk about PRF (pseudo–Relevance Feedback), which is different from the descriptor and doesn't need any more information. In this method, the top n nearest neighbors is thought to be correct and relevant results, and then the result is cleaned up. However, the main problem with this method is that the user has to write down the parameter n, which is too dependent on the database for this method to work well.

[14]Proposed a new way to classify 3D models using machine learning techniques. This method lets a 3D object be automatically categorized. Random point pairs are placed on the surface of an unclassified query instance called sq to get a sample of its category. Shape distribution histograms are made between the square and all training 3D objects by figuring out the IN, OUT, and Mixed distances between the square and all training 3D objects. When you look at a 3D object, you see a line that connects two points entirely inside the object. Out distance is the line connecting two points entirely outside of the model. Mixed distance is the line that connects two points both inside and outside of the 3D object. A machine learning technique called K-Nearest Neighbor (KNN) is then used to classify the query object by selecting the nearest K examples for classification. However, one of the major issues with this approach is that since it uses KNN algorithm, the algorithm performance depends heavily on the quality of the training datasets.

[15]Proposed a way to find 3D models based on class vocabularies (CV-3DMR) that uses category information from the classified database. It takes in the critical class information from the database. Pose normalization is done only for positions and scales as part of the algorithm. This means that pose normalization for rotation isn't done at this point. Then, multi-view rendering is when depth buffer images of the 3D object are rendered from 42 angles on a view sphere that is all the same. Next, a scale-invariant feature transform (SIFT) algorithm is used to find visual features close to where you are. SIFT features of 3D objects from all classes are used to train the universal vocabulary. Then, class vocabularies are changed using data from their classes with the maximum a posteriori (MAP). It then encodes the 3D object's local SIFT features with the Fisher Kernel and the correct vocabulary so they can be found later. A distance-vector revision strategy is used to improve search results based on the top results for the main query. One of the main problems with this method is that the precision of the retrieval list is somewhat unbalanced

2.3 Unsupervised Algorithms with No Additional Knowledge

[16] Proposed a novel composite feature vector called multi-Fourier spectra description (MFSD), composed of four different spectra via individual four Fourier transformations with periphery enhancement, augmented with spectral clustering greatly enhance the retrieval process. The Fourier transformations include depth buffer, silhouette, contour, and voxel transformations. The first step of the MFSD feature vector computation is the pose normalization which means adjusting the size, location, and orientation of a given 3D object in canonical space. MFSD feature vector computation can be achieved by Point SVD, based on Principal Component Analysis (PCA). In PCA, the object's principal axes are computed from the collection of random points on the surface of a given 3D object or by computing surface regular associated with the random point on the 3d object called Normal SVD. Feature vector from the depth-buffer image, silhouette image, contour image, and voxel are combined to form the composite feature called MFSD. Finally, as mentioned above, MSFD is augmented with spectral clustering to enhance the retrieval process. A p-minimum spanning tree (p-MST) is defined among 3D object data sets in spectral clustering. The affinity matrix is computed from the p-MST followed by dimensionality reduction from the entire feature vector space to k (the number of clusters in the spectral clustering) dimensional space. However, the major drawback of this method is that the clustering of the whole database is not parameter-free. They use a k-means clustering in spectral space and generate a single clustering. The number of clusters needs to be chosen along with other parameters dependent on the database.

[1]Proposed a method of retrieving unsupervised 3D objects by using a PANORAMA descriptor augmented with Hierarchical clustering (HC). This method doesn't require additional classification information as most

Volume: 14 Issue: 03 Pages: 5455-5463(2022) ISSN: 0975-0290

methods do. However, the method suffers from the additional runtime of the HC.

[17] Proposed a various future selection methods such as information gain, gain ratio and correlation based feature selection. In their paper, they select 33 features out of 41 for classification and the results for various classifiers are compared. Where the sample cart algorithm gives the highest accuracy 66.77% whereas the classification result of C4.5 decision tree is 65.65% only.

3. MATERIALS AND METHODS

This section describes the proposed IP3DOR system algorithm. The system consists of a database of 3D objects. We used Princeton Shape Benchmark (PSB) in our system and Panorama Descriptor [10]to calculate each object distance. The output of our system is a retrieval list sorted according to similarity with the query object. The algorithm uses a recomputed hierarchy of clusters, and output is a retrieval list comprising all objects of the dataset ordered in decreasing order by the similarity of the user's query.
The algorithm named IP3DOR is shown below.

Algorithm: IP3DOR
Input: Princeton Shape Benchmark (PSB) datasets
Output: a retrieval list of PSB sorted according to user query object

1. **for all** $O_i \in$ Database **do**
2. *Initiate cluster* $C_i = \{O_i\}$
3. **end for**
4. **for** *i = 0 to n-2* **do**
5. *Compute all distances* $\Delta[C_j, C_k]$
6. *Merge* C_j*, and* C_k *(j ≠ k) with smallest distance*
7. **end for**
8. **compute** the optimal hierarchical level
9. **compute** the clusters at the optimal hierarchical level
10. **foreach** *Cluster* $C \in$ Database **do**
11. **foreach** *Object* $O_i \in C$ **do**
12. **foreach** $O_j \in C$ **:** *j = i+1 to n-2* **do**
13. *compute all distance between* O_i *and other n-1 Objects* $\in C$
14. **end for**
15. *compute the average distance of* O_i
16. **end for**
17. *select Object with median average distance as representative of* C
18. **end for**
19. user queries the system by selecting a **query object**
20. **compare** user's query with each **cluster representative** ∈ Database
21. **select cluster representative** with the smallest distance to the **query object**
22. **sort all** objects in *C* in order of similarity to the **query object**
23. **retrieve all** object ∈ *C*

Line 1 to 7 of algorithm is the general algorithm of AHC from the existing scheme of [1]. Here, the tree structure of the 3D dataset (dendogram) is produced using the agglomerative hierarchical clustering. We also adopted the average linkage distance function to compute distance. We used average linkage because 3D object databases tend to have a very high variation of size and density[1].

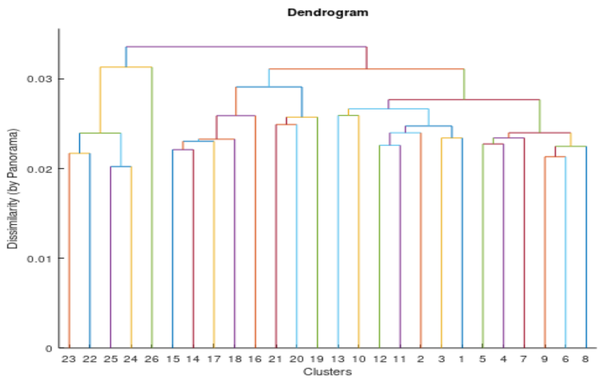


Fig.4. 1: Dendogram representing clusters and their dissimilarity

Line 8 and 9 of algorithm calculates the optimal hierarchical level and obtain the clusters contained at that level in the. Line 10 to 18 is to compute the clusters representatives of each cluster. The clusters representatives are computed using the average linkage distance function as follows[18]:
If d is a random coefficient of dissimilarity, symbols C_1, C_2 are two different clusters, Ai object belongs to a cluster C_1 and object Aj belongs to cluster C_2 then

$$d_{AL}(C_1, C_2)\ \frac{1}{n_1 n_2}\sum_{O_i \in C_1}\sum_{O_j \in C_2} d(O_i, O_j) \quad \text{.......... (4.1)}$$

Determines the distance of clusters for the average linkage method, where n1 and n2 are the numbers of objects in clusters C1 and C2. Line 1 to 18 in the algorithm is all computed offline. This means the clustering and cluster representative computations are done before the user queries the system. This will eliminate the additional run time of hierarchical clustering. Line 19 in the algorithm signifies the beginning of the online transaction. This is where the user chooses a query 3D object from the system to query the system. We assume that the query object is always contained in the dataset. Line 20 in the algorithm compares the query with each cluster representative. That is to calculate the dissimilarity between the query and each cluster's representatives. In Line 21, the cluster representative with the smallest dissimilarity is considered, and then 3D objects are retrieved in order of similarity to the query from that cluster at lines 22 to 23. Our contribution to the existing algorithm is introducing cluster representative for each cluster, thereby reducing the turnaround time of the search operation.

4. EXPERIMENTS AND RESULTS

This section discusses the experimental setup to evaluate our proposed IP3DOR system. We evaluate the system's effectiveness by measuring the retrieved list's precision, recall, and F-measure. However, we evaluate the system's efficiency using the online execution time.

Volume: 14 Issue: 03 Pages: 5455-5463(2022) ISSN: 0975-0290

4.1 Experimental setup

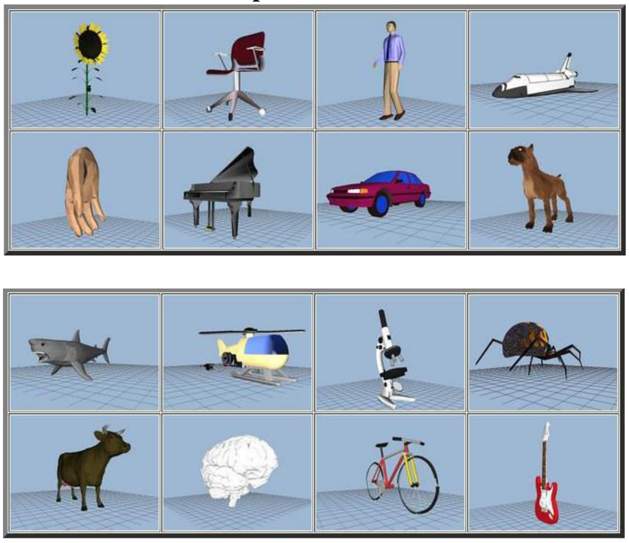
We used computer intel ® core i5- 4310 2.6GHz with 8GB of memory running windows ten pro. We implemented the proposed system in octave 6.2.0

4.2 Data set

Princeton Shape Benchmark (PSB) [19]is a collection of three-dimensional models culled from the Internet. It comes with a dataset of 1814 generic three-dimensional models saved in the Object File Format (. off). The benchmark models have been divided into two databases: one for training and the other for testing.

Princeton Shape Benchmark dataset [19]was used in the two experiments of [1] and IP3DOR. We combined the training and testing datasets as done on [1]. PSB comprises three-dimensional models with class keywords. In our experiments, we took each of the models in the dataset as a user query. When the class keyword matches an object's cluster, that object is said to be correctly placed in the retrieval list.

Table1: **Princeton Shape Benchmark**



4.3 Shape Descriptor

This research will use the PANORAMA descriptor to calculate the distance between two three-dimensional objects. The panorama shape descriptor makes use of a collection of panoramic views acquired by translating the three-dimensional object to all of the sides of the object, excluding its base and top that is oriented with the object's principal axis. It is a state-of-the-art shape descriptor that was developed by [10].

4.4 Evaluation Metrics

The performance indicators to analyze the proposed approach are the precision-recall, E & F measure (E-M & F-M), and online execution time. Precision-recall and E & F measure the system's effectiveness: the similarity between the set of 3D models retrieved by the system and the set of relevant 3D objects provided by specialists. Execution time tends to measure the efficiency of these algorithms; that is, the time is taken to retrieve the list. Efficiency here means how economically the system is achieving its objectives, retrieving relevant 3D models from the dataset. At the same time, effectiveness is the level to which our system can achieve its objective of retrieving relevant models while withholding non-relevant models.

4.4.1 Precision-recall

In evaluating three-dimensional retrieval systems, the precision-recall graph metric is the most often used measure. According to [20], "precision is the ratio of retrieved objects relevant to all retrieved objects in the ranked list, while recall is the ratio of relevant objects retrieved in the ranked list to all relevant objects in the dataset."

Assuming a dataset contains n objects where n is a positive integer, precision (P) assesses the accuracy of the relevant objects among the top n ranking results. Recall (R) is to assess the proportion of the relevant class retrieved among the top n results.

Suppose A and B are the sets of all relevant and retrieved items, respectively.

$P = (A \cap B)/(B)$ ……………. (ii)

$B \neq \{\}$

and

$R = (A \cap B)/(A)$ ……………. (iii)

$A \neq \{\}$

The recall of a retrieval system measures how effectively it discovers what we want, while the algorithm's precision measures how well it filters out what we don't want. There is a trade-off between recall and precision; one may boost recall by retrieving more, but this will result in a reduction in precision [20],

4.4.2 F and E-Measures

The F and E-Measures are two measures of retrieval performance that integrate precision and recall into a single score to assess retrieval effectiveness. It is the weighted harmonic mean of precision and recall [19],it is defined as follows:

$F = 2 \times p \times r/p + r$ (iv)

F is 0 when no relevant model is retrieved and 1 when all retrieved models are relevant.

The E-Measure is defined as

$E = 1 - F$ (v)

Substituting (iv) in (v)

$E = 1 - 2pr/p + r$ ……………. (vi)

The E-measure has a maximum value of 1.0; the higher the E-measure, the better the retrieval technique. The E-measure's most crucial characteristic indicates how effectively the results were retrieved at the top of the sorted list. E-Measure is an essential evaluation metric since, generally speaking, search engine users are more interested in the first page than the subsequent pages.

4.4.3 Online Execution Time

It represents the time taken to execute an algorithm from when the user queries our system to when the user gets the retrieval list. It measures the efficiency of the algorithm. We shall adopt online execution time in

Volume: 14 Issue: 03 Pages: 5455-5463(2022) ISSN: 0975-0290

measuring the efficiency of both systems. We expect to achieve less online execution time while maintaining the same quality of retrieval lists.

4.5 Result and Discussion

[1]		Our Algorithm	
PRECISION	RECALL	PRECISION	RECALL
1	0	1	0
0.8	0.1	0.89	0.1
0.74	0.2	0.81	0.2
0.69	0.3	0.75	0.3
0.645	0.4	0.715	0.4
0.6	0.5	0.65	0.5
0.56	0.6	0.62	0.6
0.49	0.7	0.53	0.7
0.38	0.8	0.4	0.8
0.24	0.9	0.3	0.9
0.11	1	0.19	1

In this section, the result of the improved parameter-free 3D object retrieval was presented and discussed

4.5.1 Results

In this section, the [1] result was presented, and it was obtained that the result of our algorithm was better than the [1]. Table 2 presents an overall result of Precision and Recall, respectively. Princeton Shape Benchmark (PSB) dataset was used to obtain the below results.

Table 2: Precision and Recall

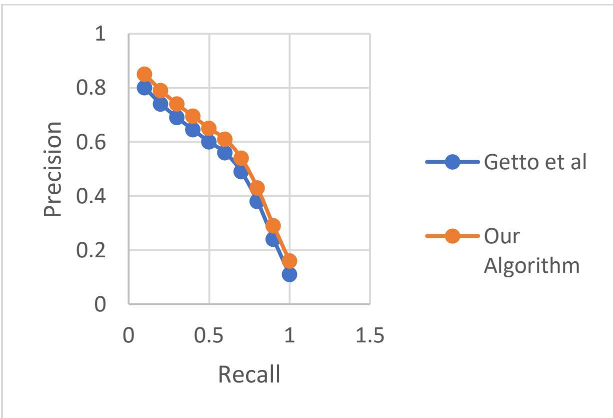


Figure 1: Precision-Recall Graph of the two approaches

From Table 2, we can compute the F & E-measure of both algorithms to compare their results further. F-measure combines precision and recall into a single number to evaluate the retrieval performance. The F-measure is the weighted harmonic mean of precision and recall. F is 0 when no relevant model is retrieved and 1 when all retrieved models are relevant. E-measure tells how good our top retrieved ranked list results are. E-measure is very important in evaluating retrieval systems since; in general, the user of a search engine is more interested in the first page of the query results than in the later pages.

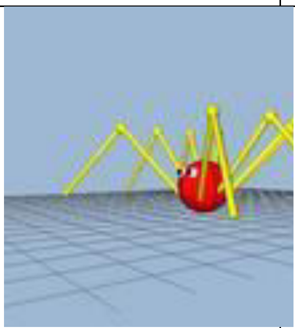
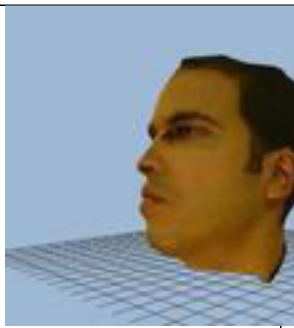
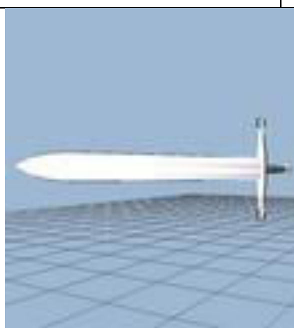
Table 3: F & E-measure of both approaches

[1]		IP3DOR	
F-MEASURE	E-MEASURE	F-MEASURE	E-MEASURE
0	1	0	1
0.1778	0.8222	0.1789	0.8211
0.3149	0.6851	0.3192	0.6808
0.4182	0.5818	0.4269	0.5731
0.4938	0.5062	0.5078	0.4922
0.5455	0.4545	0.5652	0.4348
0.5793	0.4207	0.6050	0.3950
0.5765	0.4235	0.6097	0.3903
0.5153	0.4847	0.5593	0.4407
0.3789	0.6211	0.4387	0.5613
0.1982	0.8018	0.2759	0.7241

4.5.2 Execution time

This section compares the online execution time between the [1]approach and our proposed technique and has randomly selected one query from each cluster and compared the online execution time between the two methods. We implement all retrieval methods in octave 6.2.0 on a personal computer with a 2.6GHz Intel® Core i5- 4310 CPU, 8.0GB RAM. The results reported in Table 4.3 shows that our approach significantly improves the online execution time compared to the [1]approach since, in our approach; the retrieval will be only performed in the obtained right cluster instead of systematically in the entire database. The execution time depends on the number of matched objects. Figure 4.4 shows a graph of the retrieval execution time of the two algorithms.

Volume: 14 Issue: 03 Pages: 5455-5463(2022) ISSN: 0975-0290

S/N	QUERY OBJECT	Getto et al., (2017) Retrieval Execution Time (seconds)	Our proposed method Retrieval Execution Time (seconds)
1		2.41	1.18
2		2.39	1.37
3		2.13	1.05
4		2.30	1.07
5		2.26	1.02
6		2.41	1.34
7		2.21	1.21
8		2.36	1.27

Volume: 14 Issue: 03 Pages: 5455-5463(2022) ISSN: 0975-0290

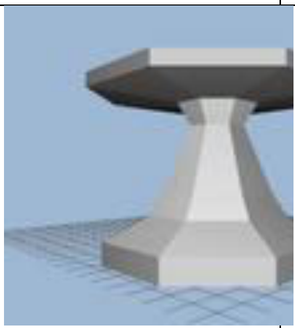
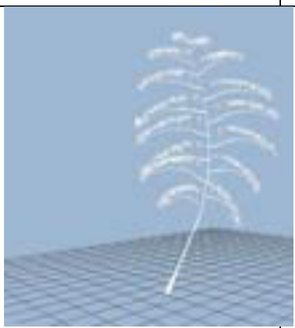
9		2.88	1.14
10		2.76	1.10
11		2.42	1.32
12		2.59	1.37

Table 4.3: Online Execution time of the two

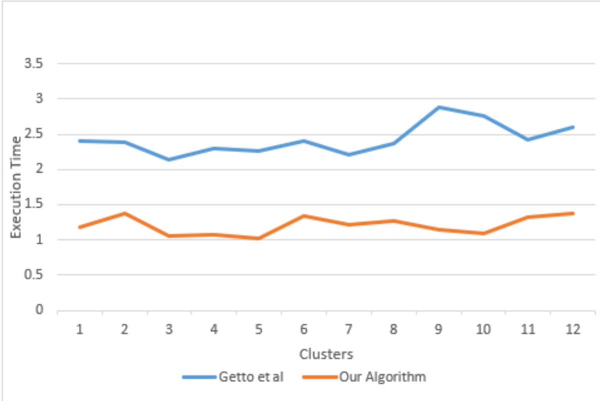


Figure 5.2: Online execution time graph of the two approaches

4.5.3 Discussion

Our technique was examined using the Princeton Shape Benchmark (PSB) dataset. It comprises three-dimensional models with class keywords. We combined the training and testing datasets in our experiment as [1]. In our experiments, we took each of the models in the dataset as a user query. When the class keyword matches an object's cluster, that object position in the retrieval list is correct. We use a Precision-recall graph in our experiment to highlight the effectiveness of the proposed algorithm regarding all user queries. The precision-recall curve shows that our algorithm improves its effectiveness. In our evaluation, we ignored the first retrieved object as this is the same with the user query. We evaluated the efficiency of our approach via the use of online execution time. Table 5.3 shows that our approach records better efficiency from the existing approach on the same machine.

In this research work, we have shown that the retrieval results of a single query can be improved by using our proposed algorithm. We introduced an algorithm that identifies and calculates each cluster's representatives to improve the efficiency and the overall retrieval performance of the existing system. The proposed system was implemented, and the system's performance was evaluated against the benchmark datasets. This revealed that the similarity measure and query search are very competitive in terms of Precision, Recall, F-measure, and retrieval execution time.

In the future, it is suggested that the technique can be improved by considering multiple representatives of the same cluster to increase our algorithm's effectiveness and efficiency and incorporating different machine learning algorithms and comparing the results.

5. CONCLUSION

In this research work, we have shown that the retrieval results of a single query can be improved by using our proposed algorithm. We introduced an algorithm that identifies and calculates each cluster's representatives to improve the efficiency and the overall retrieval performance of the existing system. The proposed system was implemented, and the system's performance was evaluated against the benchmark datasets. This revealed that the execution time was reduced by 21% and increased Precision and Recall by 30.7% and 33.1%, respectively.

REFERENCES

[1] R. Getto, A. Kuijper, and W. F. Dieter. Unsupervised 3D Object Retrieval with Parameter-Free Hierarchical Clustering. *In Proceedings of CGI 17,* 2017, 27-30 Yokohama, Japan.

[2] P. Min, M. Kazhdan, and T. Funkhouser. A Comparison of Text and Shape Matching for Retrieval of Online 3D Models. *Research and Advanced Technology for Digital Libraries* 2014, 209-220 Bath:UK

[3] B. Bostus, D. Keim, D. Saupe, and T. Schreck, Content based 3D object retrieval. *IEEE Computer Graphics and Application*, *27*(4), 2007, 22-27.

[4] Y. Gao, Q. Dai. View-based 3D object Retrieval. *Neurocomputing,* 252(C), 2015, 58-66.

[5] F. Yu,Z. Lu, and H. Luo. Three Dimensional Model Analysis and processing. China: *Springe*r 2011.

[6] H. Laga, Y. Guo, H. Tabia,R. B.Fisher, andM. Bennamoun. 3D shape Analysis Fundamentals, Theory, and Application.2019, USA: Wiley

[7] J. W. H. Tangelder, and R. C. Veltkamp. A survey of content-based 3D shape retrieval methods. *Multimedia Tools and Applications, 39*(3), 2007, 441-471

[8] B. Li, Y. Lu, C. Li. A. Godil, T. Schreck, M. Aono, and H. Johan. Shrec'14 track: extended large scale sketch-based 3D shape retrieval. *In: Eurographics Workshop on 3D Object Retrieval.* 2014,121–130. Strasbourg: France.

[9] D. Xu, and H. Li. 3D shape retrieval integrated with classification information. In Image and Graphics. *Fourth International Conference on Image and Graphics,* 2007, 774-779 Sichuan, China.

[10] P. Papadakis, I. Pratikakis, T. Trafalis, T. Theoharis, and S. Perantonis. Relevance feedback in content-based 3D Object Retrieval: A comparative study. *Computer-Aided Design and Applications, 5*(5), 2008, 753–763.

[11] D. V. Vranic.3D model retrieval. (Doctoral Dissertation). University of Leipzig, Leipzig, Germany, 2004.

[12] B. Li, and H. Johan. 3D model retrieval using hybrid features and class information. *Multimedia Tools and Applications, 62*(3), 2013, 821–846.

[13] S. Hou, K. Lou, and K. Ramani, K. SVM-based semantic clustering and retrieval of a 3D model database. *Computer-Aided Design and Applications, 2(1),*2005, 155–164.

[14] C. Ip, Y.Regli, L. W. C. Sieger, andA. Shokoufandeh. Automated learning of model classifications. *In Proceedings of the eighth ACM symposium on Solid modeling and applications* 2003, 322-327. Washington: USA.

[15] Y. Wang,Z. Liu,F. Pang, andH. Li. Boosting 3D model retrieval with class vocabularies and distance vector revision. *In TENCON 2015-2015 IEEE Region 10 Conference* 2015, 1–5 Macao: China

[16] A. Tatsuma,and M. Aono. Multi-Fourier spectra descriptor and augmentation with spectral clustering for 3D shape retrieval. *The Visual Computer 25*(8), 2009, 785–804

[17] K. Bajaj, and A. Arora, Improving the intrusion detection using discriminative machine learning approach and improve the Time Complexity by Data Mining Feature selection methods, *International Journal of computer science, vol. 76, Aug,* 2013

[18] B. S. Everitt, S. Landau, M. Leese and D. Stahl Cluster Analysis. Wiley, London, 2011, p348.

[19] P. Shilane, P. Min, M. Kazhdan, and T. Funkhouser. The Princeton shape benchmark. *Proceedings of the Shape Modeling International* 2004, pp.167– 178, Washington: USA.

[20] A. Godil, Z. Lian, H. Dutagaci, and R. Fang, T. P.Vanamali, and C. P. Cheung. Benchmarks, performance evaluation and contests for 3D shape retrieval. Code Vectors Clustering. *IEEE Journal:* 2018, pp 7-18